THE BUILDING BLOCKS: BINARY NUMBERS AND ARITHMETIC

NUMBERS

https://xkcd.com/953/

LEARNING OBJECTIVES

- Translate between base-ten and base-two numbers
- Represent binary numbers in both octal and hexadecimal formats
- Do simple arithmetic using these representations

THE BUILDING BLOCKS: BINARY NUMBERS, BOOLEAN LOGIC, AND GATES

CHAPTER 4

WHY BINARY?

https://www.youtube.com/watch?v=Xpk67YzOn5w
(but don’t read binary numbers as in this video)

Extra explanation: https://www.youtube.com/watch?v=RrJXLdv1i74

- Nothing in the real world is exact, it is easier to "safely" distinguish two values than ten.
- Energy consumption can be reduced with only two values.
- It is easier to construct circuits that deal with two values.

(There were computers which dealt with varying values, they were analogue rather than digital computers. They were pretty cool, but harder to program.)

THE BINARY NUMBERING SYSTEM

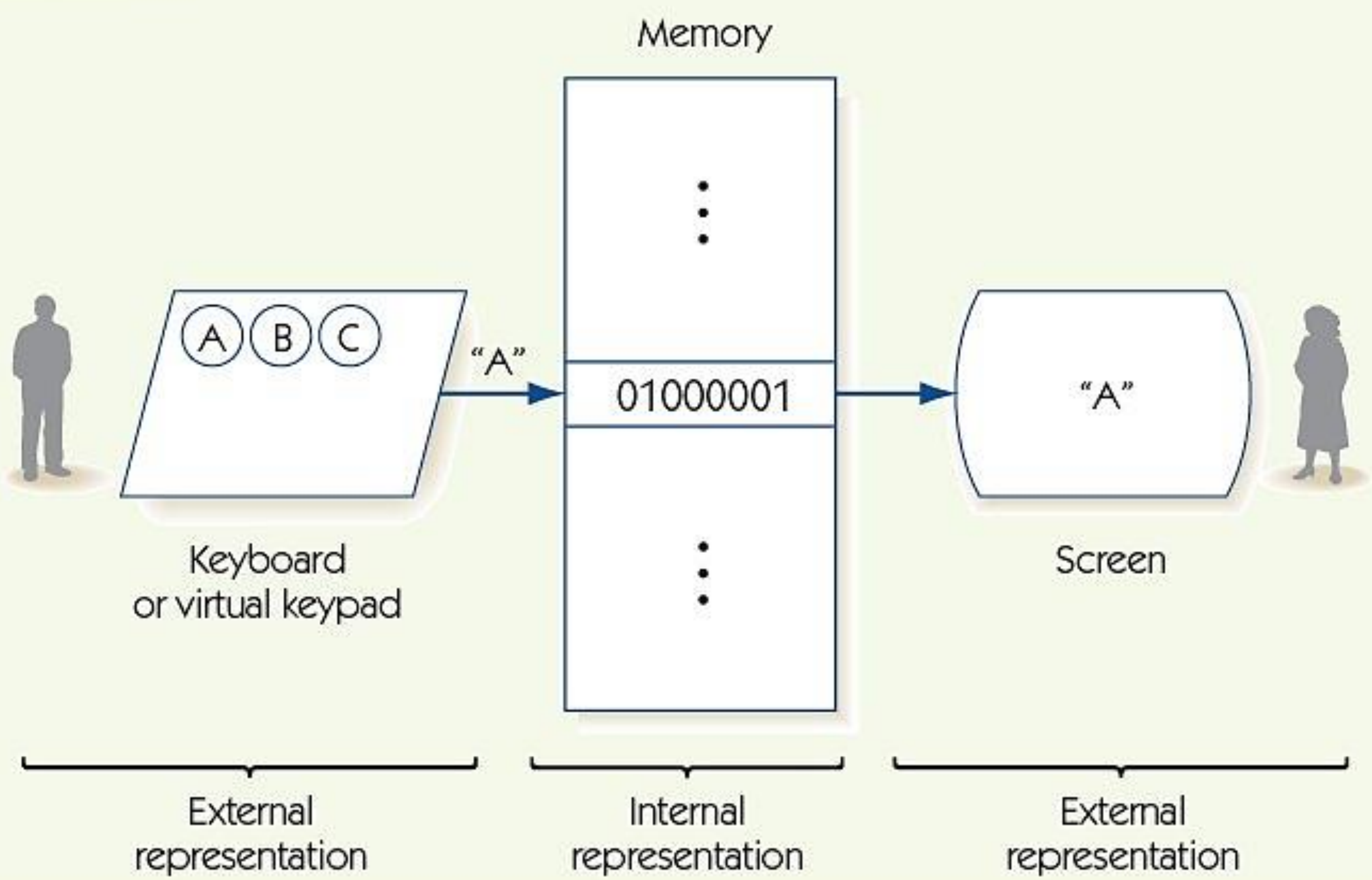
- How can an electronic (or magnetic) machine represent information?
- Key requirements: clear, unambiguous, and reliable
- External representation is human-oriented
    - Base-10 numbers
    - Keyboard characters
- Internal representation is computer-oriented
    - Base-2 numbers (binary)
    - Base-2 codes for characters

THE BINARY NUMBERING SYSTEM

- Binary is the simple idea of On/Off, Yes/No, True/False, and Positive/Negative.
- Binary is important to computing systems because of it’s stability and reliability.
    - Even when an electrical system degrades, there is still a clear “On/Off.”
- All data (and instructions) stored inside a computer are stored in binary.

THE BINARY NUMBERING SYSTEM

FIGURE 4.1



Distinction between external and internal representation of information

THE BINARY NUMBERING SYSTEM

- The **binary numbering system** is a base-2 **positional numbering system**
- Base ten:
  - Uses 10 digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9
  - Each place corresponds to a power of 10
  - $1{,}943 = (1 \times 10^3) + (9 \times 10^2) + (4 \times 10^1) + (3 \times 10^0)$
- Base two:
  - Uses 2 digits: 0 and 1
  - Each place corresponds to a power of 2
  - $1101 = (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = 13$

THE BINARY NUMBERING SYSTEM

FIGURE 4.2 Binary-to-decimal conversion table

Binary	Decimal
0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

Binary	Decimal
10000	16
10001	17
10010	18
10011	19
10100	20
10101	21
10110	22
10111	23
11000	24
11001	25
11010	26
11011	27
11100	28
11101	29
11110	30
11111	31

- https://www.youtube.com/watch?v=uotLQjvaG34

THE BINARY NUMBERING SYSTEM

- Converting from binary to decimal
  - Add up powers of two where a 1 appears in the binary number
- Converting from decimal to binary
  - Repeatedly divide by two and record the remainder
  - Example, convert 11_{10}:
    - 11 ÷ 2 = 5, remainder = 1, binary number = 1
    - 5 ÷ 2 = 2, remainder = 1, binary number = 11
    - 2 ÷ 2 = 1, remainder = 0, binary number = 011
    - 1 ÷ 2 = 0, remainder = 1, binary number = 1011

CONVERTING FROM BINARY TO DECIMAL

- You need to be able to multiply by 2 (remember no calculators in the test and exam 😀).
- So you don't need to memorise this table.
  - but I imagine you will learn these with practice anyhow

Power of 2	Decimal
2^0	1
2^1	2
2^2	4
2^3	8
2^4	16
2^5	32
2^6	64
2^7	128
2^8	256
2^9	512
2^{10}	1024 (close to 1000)
2^{11}	2048
2^{12}	4096
2^{13}	8192
2^{14}	16384
2^{15}	32768

EXAMPLES

10111_2

1	0	1	1	1	
16		4	2	1	$= 23_{10}$

Power of 2	Decimal
2^0	1
2^1	2
2^2	4
2^3	8
2^4	16
2^5	32
2^6	64
2^7	128
2^8	256
2^9	512
2^{10}	1024
2^{11}	2048
2^{12}	4096
2^{13}	8192
2^{14}	16384
2^{15}	32768

10000000000_2

1	0	0	0	0	0	0	0	0	0	0	
1024											$= 1024_{10}$

YOU DO - CONVERT TO DECIMAL

10010_2

110010_2

11110001_2

$1111\ 1111\ 1111\ 1111_2$ - frequently we write binary with gaps after every 4 digits, otherwise it becomes too hard to read

Extra for experts - what is a quicker way of doing the last one?

Power of 2	Decimal
2^0	1
2^1	2
2^2	4
2^3	8
2^4	16
2^5	32
2^6	64
2^7	128
2^8	256
2^9	512
2^{10}	1024
2^{11}	2048
2^{12}	4096
2^{13}	8192
2^{14}	16384
2^{15}	32768

CONVERTING FROM DECIMAL TO BINARY

- Every odd number has a 1 as the last binary digit
- Every even number has a 0 as the last binary digit
- We can divide the decimal by 2 and
  - if the number is even the remainder is 0 e.g. 6 / 2 = 3 r 0
  - if the number is odd the remainder is 1 e.g. 7 / 2 = 3 r 1
  - the remainder is hence the first digit of the binary
- We can keep doing this, dividing the previous answer (quotient) by two to get the next digit (dividing by 2 in binary is like dividing by 10 in decimal - you are moving one place to the right)

EXAMPLE

766

Answer so far	remainder
766	
383	0
191	1
95	1
47	1
23	1
11	1
5	1
2	1
1	0
0	1

read answer this way: 1011111110_2

EXAMPLE

12345

Answer so far	remainder
12345	
6172	1
3086	0
1543	0
771	1
385	1
192	1
96	0
48	0
24	0
12	0
6	0
3	0
1	1
0	1

read answer this way: 11000000111001_2

YOU DO - CONVERT TO BINARY

25_{10}

255_{10}

1000_{10}

32768_{10}

Power of 2	Decimal
2^0	1
2^1	2
2^2	4
2^3	8
2^4	16
2^5	32
2^6	64
2^7	128
2^8	256
2^9	512
2^{10}	1024
2^{11}	2048
2^{12}	4096
2^{13}	8192
2^{14}	16384
2^{15}	32768

Extra for experts - what is a quicker way of doing the last one?

HOW MANY BITS?

- If we want to represent an integer in binary we need a minimum number of bits.
- We can work this out by converting the number into binary and then counting the number of bits. $17_{10} = 10001_2$ so 5 bits
- Or more simply just finding the **first power of two which is bigger than the number** e.g. 17 then $32 = 2^5$ is the first power of 2 bigger than 17, so 17 requires 5 bits (what if the number was 32?)
- If we know what logarithms are we can do
  - integer (log n / log 2) + 1
  - integer (log 17 / log 2) + 1 = integer(4.0874628413) + 1 = 5

REVISION

- Convert 102 in decimal into binary.
- Check by converting back.
- How many bits do we need to represent 200 in binary?

GROUPING

- Binary is fine for computers and depending on the language you use to program with it is sometimes helpful to work in binary even for humans.
- We have already seen that we commonly group binary digits in groups of 4 starting from the right hand side to make it less likely to read it incorrectly.
- Also because reading binary is a real pain (you really should call 100_2 one-zero-zero base 2, rather than one hundred base 2) we would rather have a shorthand way of using binary.
- If we group binary digits into 4s and rename these we have hexadecimal.
- If we group binary digits into 3s and rename these we have octal.

OCTAL AND HEXADECIMAL

- Octal is really base 8 and hexadecimal (hex) is really base 16 and we can work with them in the same way as decimal and binary.
- However it is usually best just to think of them as ways of rewriting binary.

binary	octal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

because base 8 only uses the digits from 0 to 7 we can just use the ordinary decimal digits

binary	hex
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7

binary	hex
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

but we need extra digits in hexadecimal that is why A to F are also digits

OCTAL AND HEXADECIMAL

- So one way to convert to octal or hexadecimal is to convert to binary and then group in 3s or 4s to find each digit.
- e.g. 234_{10} is 11101010_2
  - if we group in 3s we get 11 101 010 which is 352_8
  - if we group in 4s we get 1110 1010 which is EA_{16}
- You do: 511_{10} is 111111111_2 what is it in base 8 and base 16?

THE BINARY NUMBERING SYSTEM

- Computers use fixed-length binary numbers for integers, e.g., 4 bits could represent 0 to 15
- In general n bits give us 2^n values, 0 to $2^n - 1$
- **Arithmetic overflow**: when the computer tries to make a number that is too large, e.g., 14 + 2 with 4 bits
- Binary addition: 0 + 0 = 0, 0 + 1 = 1, 1 + 0 = 1,

  1 + 1 = 10 (that is, 0 with a carry of 1)
- Example: 0101 + 0011 = 1000

BINARY ADDITION IS TRIVIAL

1011		1111 1101
+ 10		+ 110

BINARY SUBTRACTION IS SLIGHTLY HARDER

1011		1111 1101		1000 0000 1001
- 10		- 110		- 1111

HEX ADDITION IS SIMILAR

B		FD
+ 2		+ 6

HEX SUBTRACTION

B	FD	809
- 2	- 6	- F